

ПЛАНИРОВАНИЕ РАЗМЕЩЕНИЯ ПРОГРАММ В СИСТЕМАХ НА КРИСТАЛЛЕ

Масюков И. И., Борзов Д. Б. (ЮЗГУ, г. Курск, Россия)

Тел./Факс: +8 (960) 6789413; E-mail: ilmas46ru@gmail.com

Abstract: *In parallel organization of a computer system to become a critical criterion of the location of the modules to each other, because the data transfer rate between the modules depends on the length of the tracks connecting them. In terms of logic control systems and fault-tolerant systems, this criterion becomes even more important.*

Key words: *PLD, SoC, adjacency matrix, algorithm.*

Системы на кристалле (СнК) стали неотъемлемой частью нашей жизни. Их изучению и разработке посвящено множество различных работ. По достоинству оценено преимущество СнК, так как у нас в области разработки и создания сложной наукоемкой продукции в сохранились условия, кадры, научный потенциал. Большое число предприятий и учреждений способно разрабатывать уникальные электронные устройства. Особенно это становится актуально, когда отечественное производство не может сравниться по объемам серийного производства с западными конкурентами[1].

Разработка реконфигурируемых систем прочно связана с СнК, так как и разработка, отладка и применение связаны непосредственно с ней. При параллельной организации вычислительной системы становится критичен критерий расположения модулей между собой, так как скорость передачи данных между модулями напрямую зависит от длины трасс их соединяющих. В условиях систем логического управления и отказоустойчивых систем данный критерий становится еще более значимым. Отсюда и возникает проблема изменения внутренней конфигурации вычислительной системы. Наиболее эффективным решение данной задачи может послужить перенос вычислительной системы внутрь одной или нескольких ПЛИС. Необходимость планирования перерасположения параллельных процедур усугубляется тем, что известные методы и алгоритмы решения задачи размещения имеют большую вычислительную сложность и решаются в основном программно.

Решение задачи планирования размещения и перерасположения процедур в хост-ЭВМ сложно осуществить, так как программная реализация требует больших временных затрат. Предварительное формирование всевозможных вариантов размещения на стадии проектирования ПЛИС и организация хранения данных вариантов во внешней памяти хост-ЭВМ ведет к увеличению временных затрат. Также, кроме затрат машинного времени на получение различных вариантов размещения и ресурсов внешней памяти на их хранение, использовать ее неэффективно из-за большого времени обращения к внешней памяти ЭВМ. Все это приводит к существенному уменьшению коэффициента готовности системы.

Поэтому в рамках данной статьи рассмотрим алгоритм планирования конфигурации ПЛИС. Вся структура алгоритма построена на трех критериях[2]:

1) Модули, имеющие наибольшее количество смежностей γ должны иметь минимальное расстояние σ до смежных модулей

$$\begin{cases} \gamma \rightarrow \max \\ \sigma \rightarrow 1 \end{cases} .$$

2) Смежные модули с наибольшим количеством одинаковых инцидентных ребер λ должны располагаться таким образом, что бы расстояние между ними было минимально

$$\begin{cases} \lambda \rightarrow \max \\ \sigma \rightarrow 1 \end{cases}$$

3) Третий критерий вытекает из следующего отображения:

$$\beta_s = \left\{ \begin{array}{cccccc} x_{S_{1,1}} & x_{S_{1,2}} & \dots & x_{S_{1,k}} & \dots & x_{S_{1,n}} \\ x_{S_{2,1}} & x_{S_{2,2}} & \dots & x_{S_{2,k}} & \dots & x_{S_{2,n}} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{S_{q,1}} & x_{S_{q,1}} & \dots & x_{S_{q,k}} & \dots & x_{S_{q,n}} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{S_{n,1}} & x_{S_{n,2}} & \dots & x_{S_{n,k}} & \dots & x_{S_{n,n}} \end{array} \right\} \rightarrow \left\{ \begin{array}{cccccc} p_{1,1} & p_{1,2} & \dots & p_{1,k} & \dots & p_{1,n} \\ p_{2,1} & p_{2,2} & \dots & p_{2,k} & \dots & p_{2,n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ p_{q,1} & p_{q,2} & \dots & p_{q,k} & \dots & p_{q,n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ p_{n,1} & p_{n,2} & \dots & p_{m,k} & \dots & p_{m,n} \end{array} \right\},$$

где $S = \overline{1, N}$!. Символ \rightarrow означает отображение одной из вершин $x_{S_{q,k}} \in X$ на один из модулей $p_{q,k} \in H$. Здесь s – это номер очередной перестановки, соответствующий S -му варианту размещения. Мощность множества $\Psi = \{\beta_S\}$ всевозможных отображений равна числу всевозможных перестановок подпрограмм $x_{q,k} \in X$ в матрице M : $|\Psi| = N!$. Тогда, если Ψ – множество всевозможных отображений вида, задачу размещения, можно сформулировать как поиск отображения $\beta^* \in \Psi$, такого, что:

$$T_{\beta^*} = \min_{\Psi} \begin{cases} \max \gamma \\ \max \lambda \end{cases},$$

где \min_{Ψ} соответствует поиску минимальных суммарных межсоединений σ , для условия $\max \gamma$ и $\max \lambda$.

Критериями успешности выполнения алгоритма являются расстояния между модулями. Минимальное возможное расстояние принято за 1, за это расстояние длинна соединения между соседними блоками. Так же, для упрощения вычисления, принимается за условие многослойность платы, а длинна соединительных отверстий не учитывается.

Данные критерии позволяют составить формализованный алгоритм размещения подпрограмм в ПЛИС. Примем $x_{q,k} \in X$ - неразмещенные вершины, $x_{q,k} \in H$ - размещенные. Алгоритм состоит из следующих шагов:

1. Положить $\Lambda = \{\max_i \{x_{q,k}, x_{q,k}\}\}, x_{q,k} \in X, i = \overline{1, N}$.

Если $|\Lambda| > 1$, \rightarrow п.2,

если $|\Lambda| = 1$, \rightarrow п.3,

если $|\Lambda| = 0$, \rightarrow п.4.

2. Принять $\forall \exists |\Lambda|$:

а) $\Omega = \{ \max_i \deg(|\Lambda_j|) \}, x_{q,k} \in H, i = \overline{1, N}, j = \overline{1, \psi}, \psi = |\Lambda|,$

б) $\Omega = \{ \max_i \{ |\Lambda_j|, x_{q,k} \} \}, x_{q,k} \in H, i = \overline{1, N}, j = \overline{1, \psi}, \psi = |\Lambda|,$

в) $\Omega \rightarrow H.$

3. Принять $\forall \exists |\Lambda|$:

а) $\Omega = \{ \max_i \{ x_{q,k}, |\Lambda| \} \}, x_{q,k} \in X, i = \overline{1, N},$

б) $\Psi = \{ \max_i \deg(|\Omega_j|) \}, x_{q,k} \in H, i = \overline{1, N}, j = \overline{1, \zeta}, \zeta = |\Omega|,$

в) $\Psi = \{ \max_i \{ |\Omega_j|, x_{q,k} \} \}, x_{q,k} \in H, i = \overline{1, N}, j = \overline{1, \zeta}, \zeta = |\Omega|,$

г) $\Psi \rightarrow H.$

4. Положить $G = \langle H, M \rangle.$

Можно заметить, что алгоритм сводится к поочередному выполнению похожих шагов и является линейным. Данный алгоритм может быть реализован на существующих ПЛИС с подходящей логической емкостью. Матрица межсоединений будет поступать с хост-Микроконтроллера, включающего в себя функции выбора и пересылки данных на вход ПЛИС по интерфейсу данных.

Можно заметить, что используя связку малопроизводительного микроконтроллера и ПЛИС можно добиться не только существенной экономии ресурсов – уменьшения размеров, мощности, стоимости, но и самое главное увеличения производительности, так как коммутационные задержки в ПЛИС, для реализуемого алгоритма, не высоки, а сама скорость работы ПЛИС в данном случае относительно ЭВМ выше. Для примера, на ПК с Core i7 2-ого поколения расчет матрицы размерностью 2x3 по данному алгоритму занимает приблизительно около 0.1 секунды для 10 шагов алгоритма.

В дальнейших работах будет разработана функциональная схема описанного алгоритма и будет проведено исследование сравнения скорости его работы на ПК и ПЛИС.

Список литературы: 1. Воеводин В. В., Воеводин Вл. В. Параллельные вычисления. БХВ-Петербург 2004 г. 2. Машиностроение и техносфера XXI века // Сборник трудов XXII международной научно-технической конференции в г. Севастополе 14-19 сентября 2015 г. – Донецк: МСМ, 2015. Т. 2. – 259 с.